

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>01 OCT 2014</b>		2. REPORT TYPE <b>N/A</b>		3. DATES COVERED <b>-</b>	
4. TITLE AND SUBTITLE <b>Probabilistic Analysis of Time Sensitive Systems</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release, distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>The original document contains color images.</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>SAR</b>	18. NUMBER OF PAGES <b>1</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

# Probabilistic Analysis of Time Sensitive Systems

## Problem Statement

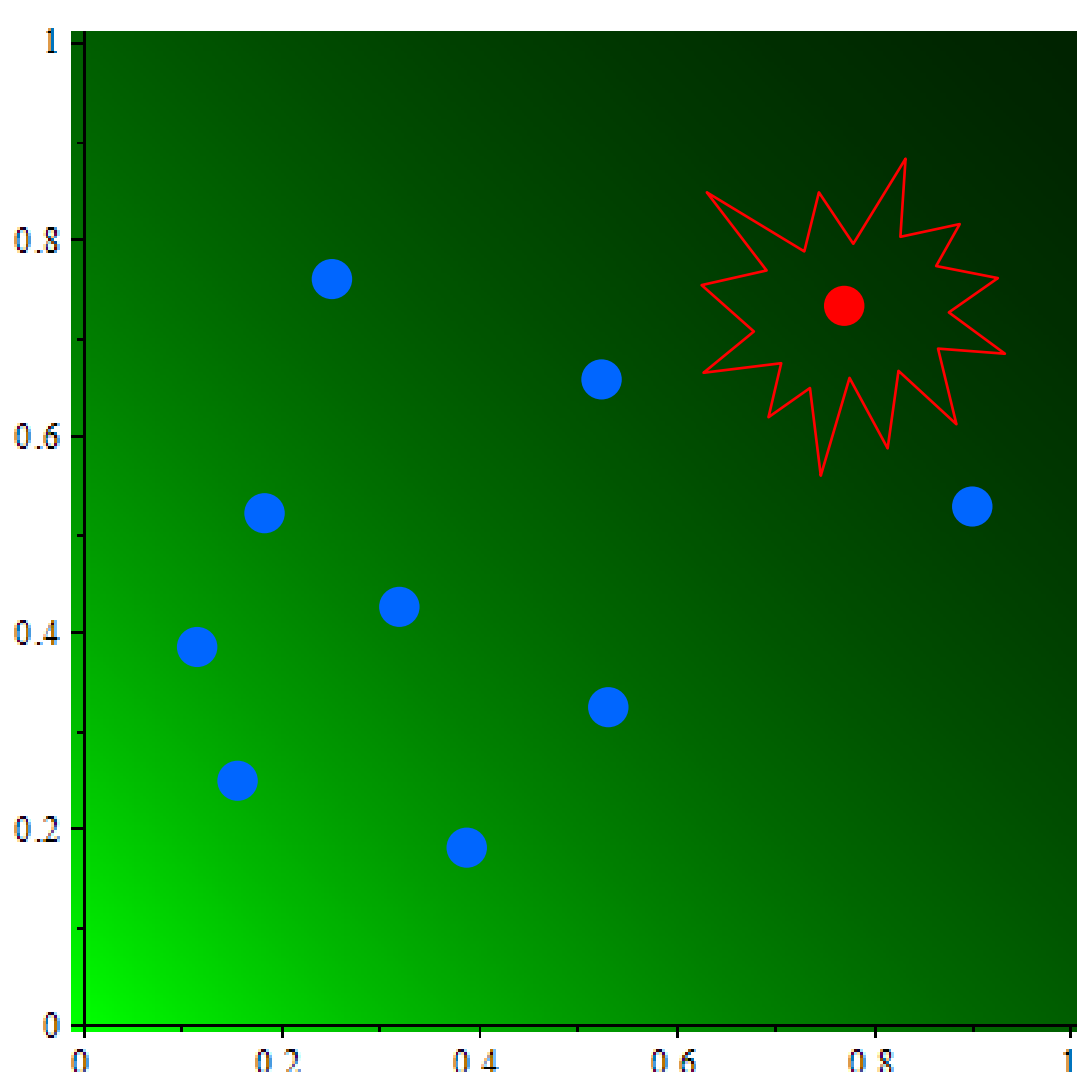
Time-sensitive systems in uncertain environments have complex behaviors. How do we assure correctness of such systems?

- Exact probabilistic verification is infeasible due to model size
- Black box testing does not yield bounded predictions
- Need formal approach for dealing with uncertainty
- Accurate, bounded, probabilistic results
- In reasonable time even for rarely occurring errors

## Stochastic Model Checking (SMC)

SMC is a rigorous simulation-based approach for estimating that a property holds in a system.

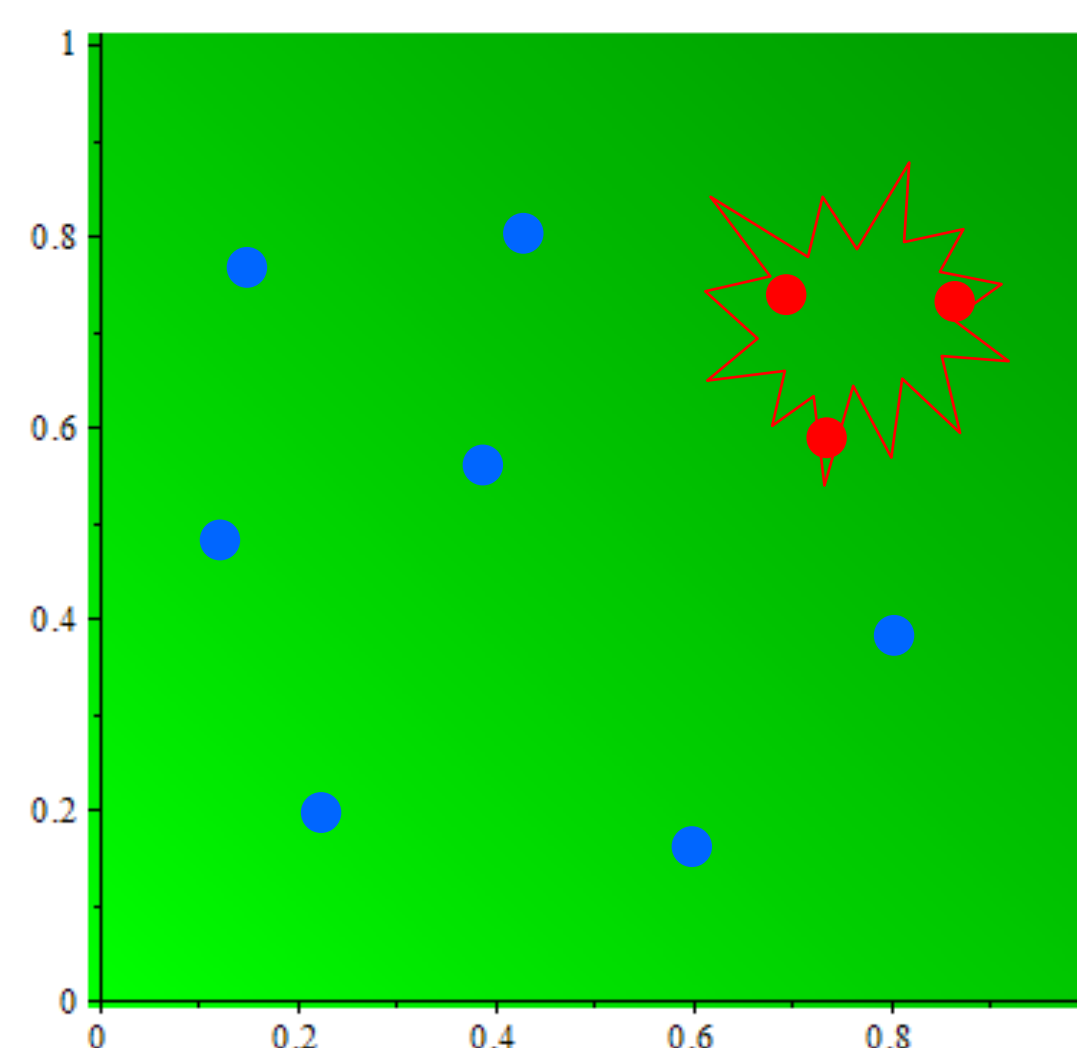
- System properties described in formal language (BLTL, etc.)
- Property is tested on “sample trajectories” (sequence of states)
- Each outcome treated as a Bernoulli trial (i.e., coin flip)



### SMC Basics

- Indicator function  $I(\vec{x}) = 1$  iff property holds for input  $\vec{x}$ .
- Relative Error  $RE(\hat{p}) = \frac{\sqrt{var(\hat{p})}}{E[\hat{p}]}$  is measure of accuracy.
- Draw random samples from input distribution  $f(\vec{x})$  until target Relative Error is met.
- Estimated probability that property holds is:

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N I(\vec{x}_i) = \frac{1}{10} = 0.1 \quad RE(\hat{p}) = \frac{0.32}{0.1} = 3.2$$



### Importance Sampling

- Modify input distribution to make rare properties more visible.
- Weighting function  $W(\vec{x})$  maps solution back to original problem.
- Reduced relative error with same number of samples.

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N I(\vec{x}_i) W(\vec{x}_i) = \frac{0.2 + 0.5 + 0.3}{10} = 0.1$$

$$\widehat{RE} = \frac{0.18}{0.1} = 1.8$$

## Semantic Importance Sampling A New Approach to Importance Sampling

### Input Specification in C

```
#include "osmosis_client.h"

//@dist a=uniform(min=0,max=5)
//@dist b=normal(mean=3,std=1,min=0,max=5)
void simple()
{
    double a = INPUT_D("a");
    double b = INPUT_D("b");
    double c = a + b;
    double d = (a - b)/2.0;

    ASSERT(sin(c)*cos(d/2) < 0.995);
}
```

Translate C  
model to SMT2  
for Analysis.



Recursively invoke  
dReal SMT checker to  
build abstract model  
of specification.

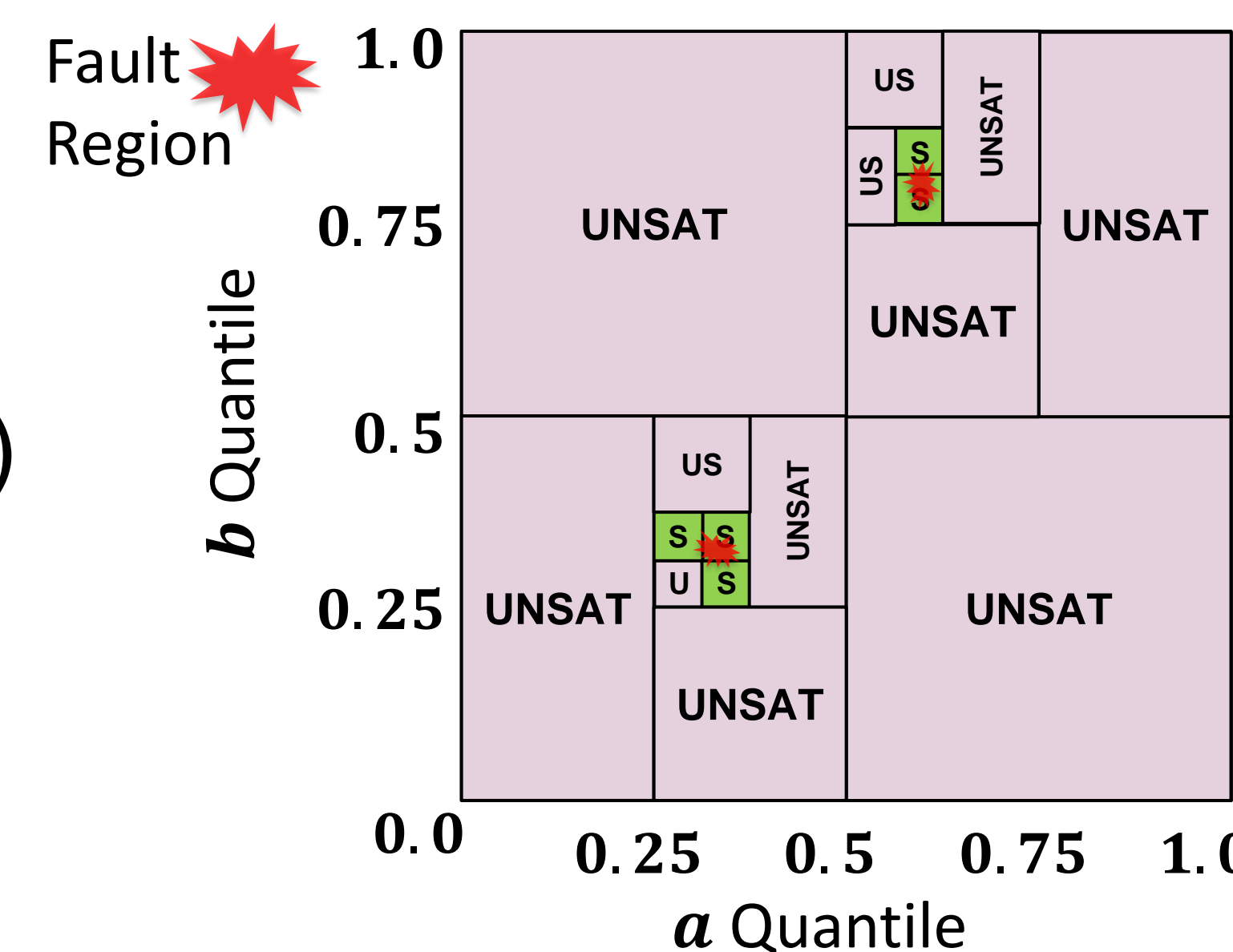
### SMT2 Model

```
(set-logic QF_NRA)
(declare-fun a () Real)
(declare-fun b () Real)
(declare-fun a_1 () Real)
(declare-fun b_1 () Real)
(declare-fun c_1 () Real)
(declare-fun d_1 () Real)
(assert (>= a 0))
(assert (<= a 5))
(assert (>= b 0))
(assert (<= b 5))
(assert (= a_1 a))
(assert (= b_1 b))
(assert (= c_1 (+ a_1 b_1)))
(assert (= d_1 (/ (- a_1 b_1) 2.0)))
(assert (not (< (* (sin c_1) (cos d_1)) 0.995)))
(check-sat)
(exit)
```

Input  
Cube

ASSERT()

### Abstract Indicator Function $I^*(\vec{x})$



Weight function  $W(\vec{x}_i)$   
is probability  $p^*$  that  $\vec{x}$   
is in  $I^*(\vec{x})$ .

### Abstract Probability

$$p^* = \frac{5}{2^8}$$

Number of cubes in  $I^*(\vec{x})$ .

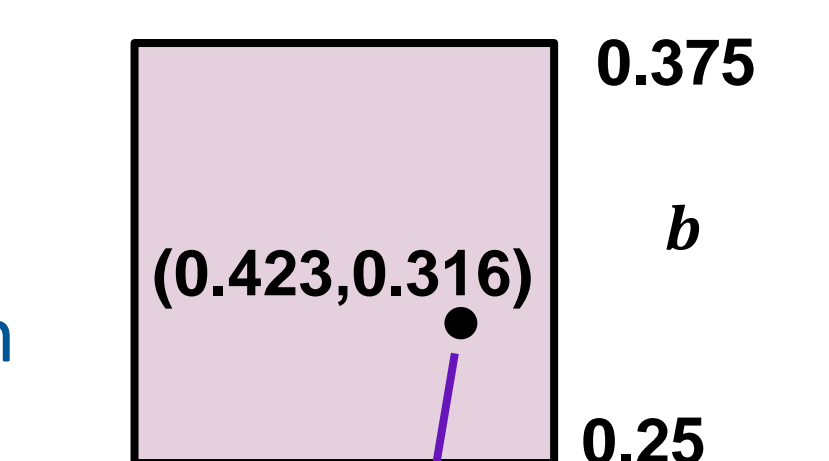
Level of cubes

### Raw Prob. Estimate

$$\hat{p}_{raw} = 0.024$$

$$RE(\hat{p}_{raw}) = 0.01$$

### Input Generation



Use  $I^*(\vec{x})$  to generate random  
input vectors:

- Randomly pick SAT cube
- Randomly pick point in cube

Apply inverse CDF on  
each input variable.

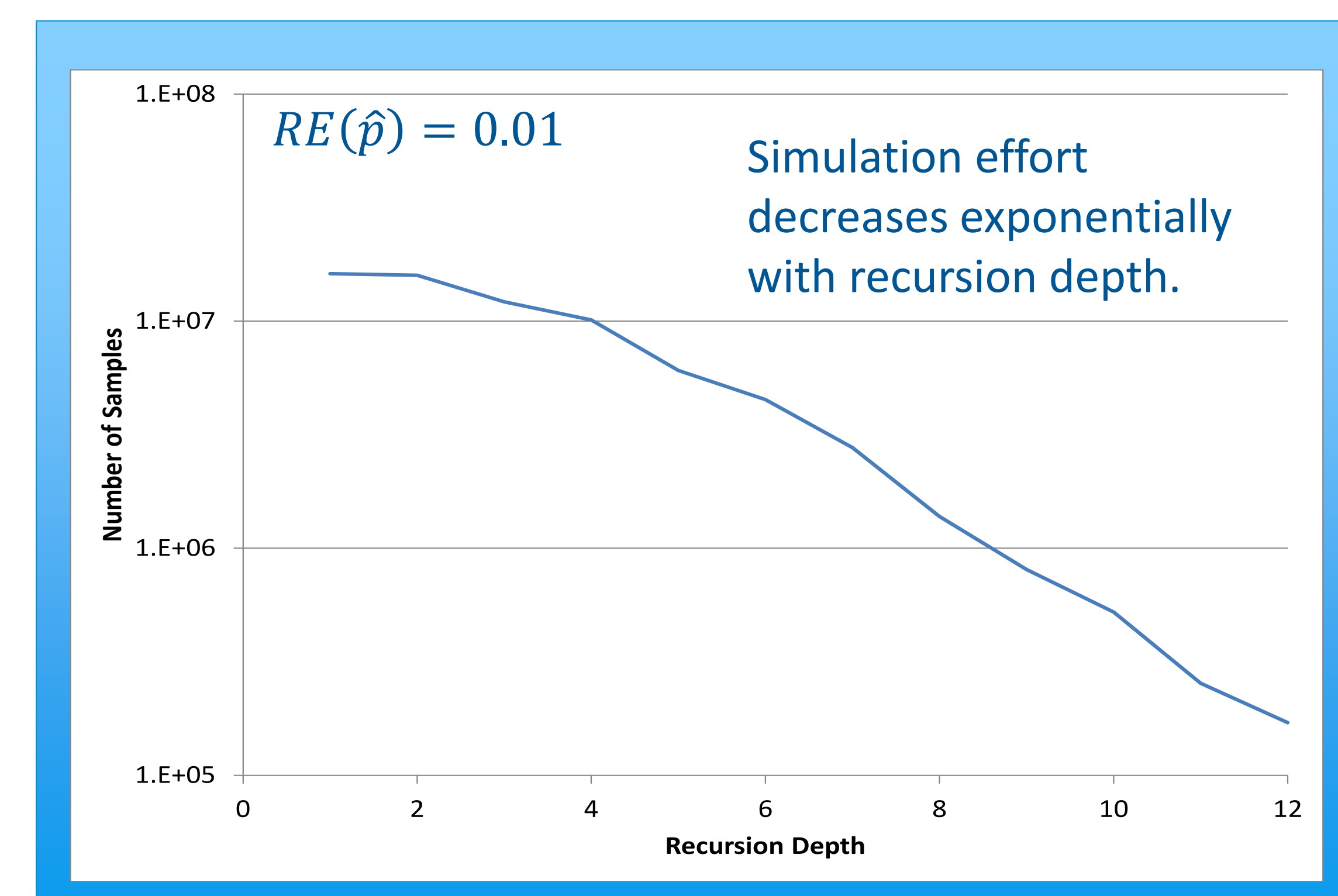
$$(a, b) = (2.115, 2.503)$$

Apply generated inputs to  
original C model to calculate  
bounded failure probability  
estimate.

### Final Probability Estimate

$$\hat{p} = p^* \hat{p}_{raw} = 0.00047$$

$$RE(\hat{p}) = 0.01$$





Copyright 2014 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:\* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:\* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

\* These restrictions do not apply to U.S. government entities.

DM-0001712